On the performance of group key management protocols in MANETs

Mohamed Salah Bouassida, Mohamed Bouali

On the Performance of Group Key Management Protocols in MANETs

Mohamed Salah Bouassida, Mohamed Bouali

MADYNES - LORIA, campus scientifique, BP 239, 54506 Vandoeuvre lès Nancy, France

The establishment of Group Key Management Protocols (GKMPs) is the most suitable solution to provide secure multicast communication within ad hoc networks. However, these protocols have to take into account the characteristics of a such environment, with mobility of nodes, wireless links and low capacities. The evaluation of these protocols becomes thus mandatory to guarantee their adequacy and their applicability within MANETs.
In this paper, we present an evaluation method for group key management protocols within ad hoc networks based on NS2, illustrated by the evaluation and the comparison of four main existing GKMPs: GKMPAN, DMGSA, BALADE and Hi-GDH, which belong to different group key management protocols approaches.

KeyWords: Group Key Management, Multicast, Ad Hoc, Simulations

1 Introduction

The combination of an ad hoc environment with multicast services, induces new challenges towards the security infrastructure to enable acceptance and wide deployment of multicast communication. Indeed, several sensitive applications based on multicast communications have to be secured within ad hoc environments. We can cite military applications such as group communications in a battle field, but also public security operations, involving fire brigades and policemen. To prevent attacks and eavesdropping, authentication, data integrity and confidentiality services need to be provided. The most suitable solution to offer these services is the establishment of a key management protocol. This protocol is responsible for the generation and the distribution of the traffic encryption key (TEK) to all group members. This key is used by the source to encrypt multicast data and by the receivers to decrypt it.

The adaptation and the effective applicability of group key management protocols in the context of ad hoc networks should be demonstrated, according to the constraints and the limitations of this environment. In this paper, we analyze the performances evaluation of four main GKMPs defined in MANETs using the NS2 network simulator.

To present our contribution, we structure this paper as follows. Section 2 presents a brief state of the art of the group key management protocols within ad hoc networks. In section 3, we define our simulation environment with NS2. In section 4, we discuss and analyze our simulations and results. Section 5 concludes this paper.

2 Group key Management protocols in MANETs

We classify group key management protocols into three approaches: centralized, distributed and decentralized. Figure 1 illustrates this taxonomy. In the following, we present this taxonomy and illustrate each approach by a protocol example, that we evaluate its performances in section 4.

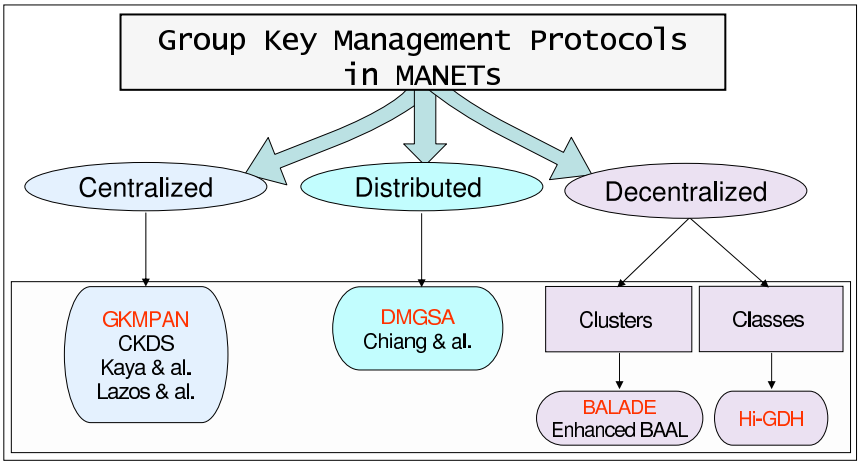

Fig. 1: Group Key Management Protocols in MANETs

2.1 Centralized Approach

In this approach, group key management is carried out by a single entity in the network. Some protocols belonging to this category do not need an off-line key pre-distribution phase like the one defined by KAYA AND AL. [KLNY03] or the one defined by LAZOS AND AL. [LP03]. Other centralized group key management protocols such as GKMPAN [ZSXJ04] or CKDS [MME04] operate with key pre-deployment phase on each node participating to the multicast group. Group members will be able to decrypt the multicast flow sent by the source, or to obtain the traffic encryption key when the re-keying process is triggered. Key pre-distribution is used in MANETs because of the lack of infrastructure within ad hoc network which implies the unavailability of a central entity to ensure key distribution initialization on-line. The GKMPAN protocol [ZSXJ04] is based on a key lists pre-distribution phase to the multicast group members and on multiple rekeying phases. The main phases of this protocol are the following:

- Key pre-distribution: each group node u obtain, off-line, before the deployment of the ad hoc network, a subset I_u of m keys out of the pool of l keys. These keys are used as key encryption keys (KEKs). The key-predistribution algorithm allows any node who knows another node's identifier j to determine the identifiers of I_j.

- Authenticated node revocation: when the key server decides to revoke a node, it broadcasts a revocation notification to the network, containing the identifier of the revocated node, and the non compromised key that is possessed by the maximum number of remaining nodes in the network.

- Secure group key distribution: The key server generates and distributes a new group key. The key distribution process is achieved hop by hop, by encrypting the new group key with the predeployed KEKs. When a node is compromised and is revocated by the key server, its predeployed KEKs are also compromised. To face this problem when sending the new group key, the key server determines the identifier of the non compromised KEK, shared with the maximum members of the multicast group. Then, it broadcasts a message authenticated by TESLA, containing the new group key encrypted with this chosen non compromised KEK. Group nodes who did not hold the KEK used for the encryption of the traffic encryption key, will receive this group key, forwarded by their neighbors, encrypted with other non-compromised KEKs. So, the key server has only to deliver the new group key to its immediate neighbors, which forward it securely to their neighbors, in a hop by hop manner. Thus, GKMPAN exploits the multi-hop property of the ad hoc networks, group members are both hosts and routers.

- Key update: When the group nodes decrypt and authenticate the traffic encryption key, they update their subsets of predeployed KEKs, based on this group key, and erase all the old KEKs. The compromised keys k_i are also updated by the remaining members holding these keys, using a non compromised key k_m as follows: $k'_i = f_{k_m}(k_i)$. f being a pseudo-random function.

2.2 Distributed Approach

Group key management in the distributed approach is achieved by all the multicast group members, which cooperate to ensure a secure multicast communications between them. The protocols proposed by CHIANG ET AL. [CH03] and DMGSA [KLG06] illustrate this approach.

DMGSA [KLG06] (Distributed Multicast Group Security Architecture) is a distributed and clusterized security architecture. It takes into account the mobility and density of group nodes when the clusters formation of the multicast group. The group key management is realized via a special entities in the network, called GCKSs (Group Control Key Server). These entities are the clusterheads and form together the backbone of the multicast group. Thus, there is no centralized point of vulnerability in the network. In each k-hop neighborhood, a GCKS is elected each time a modification or change of the topology occurs. The GCKS election is realized in a distributed manner, following two steps: the clusters formation phase and the clusters maintenance.

The distributed clusters formation process is initiated by a node which does not belong yet to a cluster. This node diffuses election messages, in its k-hops neighborhood, to claim itself as a clusterhead (GCKS). The choice of k is based on the estimation of the node of the local density of its neighborhood. This estimation is computed via a neighborhood detection algorithm. In case of competition between two nodes, the node holding the smallest value of k and the smallest identifer is elected as GCKS.

During the clusters maintenance phase, each clusterhead sends periodically a notification message to claim itself as the GCKS in its k-hops neighborhood. This message maintain the members which receive it belonging to the cluster. A member which does not receive the notification message from its GCKS, during a defined period of time, will join another cluster. Key management in DMGSA consists on sharing a group key TEK, managed by the GCKS group. Each member of the multicast group receives from its GCKS the TEK. To distribute the TEK in a secured manner, the GCKS authenticates their local members when they join the group, and verify if they are authorized to accede to the multicast flow. This access control is done through certificates deployed off-line. In case of success, the GCKS establishes with each member in its sub-group a secret key, called KEK (Key Encryption Key), used to encrypt the TEK. The TEK renewal is triggered when the adhesion frequency to a cluster (Join and Leave events), evaluated by its GCKS, exceeds a defined threshold. In this case, the GCKS generates a new TEK, sends it to its local members individually encrypted with their KEKs and to the GCKS of the multicast group.

2.3 Decentralized Approach

The decentralized approach divides the multicast group into sub-groups or clusters with same level, or into classes with different levels of group members.

In a cluster approach such Enhanced BAAL [BCF04] and BALADE [BCF06a], each sub-group is managed by a local controller responsible for the security management of the members of its sub-group. In the BALADE protocol, the multicast flow is encrypted by the source using the TEK (Traffic Encryption Key), and forwarded to the group members through the established multicast tree. The architecture of BALADE is composed of three specific nodes: the global controller which is represented by the source responsible for the generation and the distribution of the TEK, the local controllers responsible for the secure TEK forwarding and the group members.The basic idea of BALADE is to clusterize the multicast group, dynamically into sub-groups via the OMCT [BCF05] algorithm. Each sub-group is managed by a local controller which shares with its local members a local cluster key. The local controllers form a multicast group and share a key encryption key, in order to ensure a secure distribution of the TEK. There is no intermediate encryption and decryption operations of the multicast flow by local clusters controllers. This is an important advantage in ad hoc networks where resources are limited.

In a class-oriented or hierarchical approach, the sub-dividing consists in assembling group members into classes, with different levels. A class level represents the function or the degree of its members. The

classification is thus closely related to the type of the application to be secured. Within a class, a controller responsible for the management and the control of its class members is elected. Each class c holds a traffic encryption key TEK_c, generated by the class controller or by the global controller of the group (who is the controller of the first group class). According to the security policies of the group, members of class i should be able to decrypt only communications of their class, or of the lower classes. The key management within the group must take into account this assumption, to provide an efficient keys distribution process, in terms of bandwidth and computation power. In this section, we present the hierarchical group key management protocol Hi-GDH. This protocol is dedicated to operate within PMR (Private Mobile Radio) networks, which allow users groups using wireless devices, to secure their voice, data or multimedia communications. This group key management protocol was designed in the context of the RNRT SAFECAST project[†].

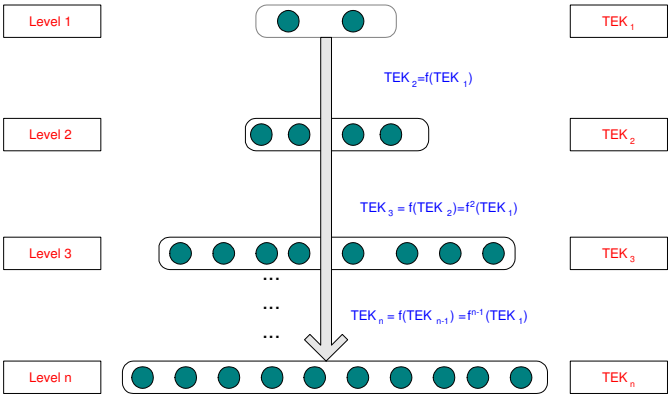

Fig. 2: Keys management in Hi-GDH

A group is sub-divided into several hierarchical classes, with different levels. as illustrated in Figure 2. Members of class i use, to ensure secure communications between themselves, a shared key called TEK_i. Among these members, a privileged agent is distinguished: the chief of the class. The chief of the group is thus represented by the chief of the first class. Members of class i have access to the secure communications of the lower classes (of level j such that $j <= i$). However, they cannot access to the communications of the upper classes. To do so, the different keys TEK_i must be linked via a one-way hash-function f: $f(TEK_i) = TEK_{i+1}$; $f^{n-1}(TEK_1) = TEK_n$ (cf. Figure 2), as follows: The keys confidentiality is ensured via the establishment of several sub-protocols, we present below the TEK generation and distribution protocol. Each member of class C_c holds a pre-deployed key TEK_c^{init} (obtained for example at its authentication). c represents the hierarchical level (C_1 is the highest level class and C_n the lowest level class). Nodes in class C_c know the keys TEK_l^{init} $(l >= c)$. The TEK generation and distribution process is carried out as follows:

1. Nodes of C_1 execute Diffie-Hellman (DH) [DH76] to ensure cooperation and collaboration between them.

2. The last node M in DH computes the TEKs via the relation $TEK_{i+1} = f(TEK_i)$.

3. M distributes the TEK_MESSAGE:
   $(\{TEK_1\}_{TEK_1^{init}}, \ldots, \{TEK_n\}_{TEK_n^{init}})$

Nodes of the class C_1 execute DH to generate TEK_1 of level 1. Then, the other TEKs of lower classes will be computed from TEK_1, thanks to the one-way function f. Therefore, a member from a level c is able to access data exchanged between members of its class and of lower classes, and is not able to reach upper levels communications.

[†] http://www.telecom.gouv.fr/rnrt/rnrt/projects/ safecast.htm

The algorithm executed by a node belonging to the class C_c is the following:

While receiving TEK_MESSAGE($m_1, m_2, ..., m_n$) with $m_i = \{TEK_i\}_{TEK_i^{init}}$, i=1,2,...,n:
BEGIN
Decrypt m_c with TEK_c^{init}
END

3 Simulation Environment

In this section, we define the the NS2[‡] simulation environment we used, a discrete event network simulator.

3.1 Simulation Metrics

We adopted three simulations metrics to evaluate the performances of group key management protocols within MANETs: the average delay, the energy consumption and the keys delivery rate.

- The average delay of keys transmission (**D**) from the source to the receivers allows to evaluate the necessary time to transmit the group key to the group members. To ensure an efficient synchronization between the encryption and decryption of the multicast flow, this delay should be optimized.
- The energy consumption (**E**) is defined as the sum of units required to the keys transmission throughout the duration of simulation. The evaluation of this metric is mandatory in our study, because the optimization of the energy consumption is a true challenge in ad hoc networks.
- The keys delivery rate (**KDR**) is defined as the ratio between the number of received keys and the number of sent keys multiplied by the number of receivers. This metric allows to measure the reliability rate of the studied protocol in terms of keys transmission to the group members.

$$KDR = \frac{Received\_keys\_Number}{Sent\_keys\_Number * Receivers\_Number}$$

3.2 Simulation Parameters

The objective of our simulations is to compare BALADE, DMGSA, GKMPAN and Hi-GDH, according to the three metrics presented above (keys transmission delay, energy consumption and keys delivery rate). The impact of the dynamicity of the group members and the impact of the relative speeds between members in a same cluster are also evaluated within these simulation. Our simulation environment is defined by the following parameters. The density of group members within the ad hoc network: group members number (30 - 40 - 50) and network surface (500m*500m, 1000m*1000m, 1500m*1500m). The mobility scenarios are generated by the automatic generator *setdest* provided by NS2; the maximal speed of members is defined at 10km/h (2.77m/sec), the pause time is 20 seconds and the simulation duration is 2000 seconds. Each simulation with a given configuration parameters is repeated 10 times. The multicast routing protocol used is MAODV. Only keys distribution traffic exist within our simulations ; it starts at 1000 second and consists in distributing the group key and renewing it every 200 second. And finally, all group members are members of the multicast flow, from the beginning of the simulation.

3.3 Presentation of the Protocols Agents in NS2

In order to compare the performances of the four protocols, we implemented four agents corresponding to these protocols and we integrated them in the NS2 simulator. The GKMPAN agent simulates a group chief which distributes the encryption key to the group members, with a centralized manner, every 200 seconds. This key distribution process being realized hop by hop, a delay is added to the packets of group key distribution by the intermediary nodes. This delay corresponds to the average transmission delay of

[‡] http://www.isi.edu/nsnam/ns/ns-build.html

keys between a member and its neighbors and to the symmetric encryption and decryption operations of keys while transmitting them.

For the DMGSA agent, the clusterheads are elected and ensure the generation and the distribution of the group key to their local members, every 200 seconds. The clusters are formed at 2-hops of their GCKS and are maintained via notification messages periodically sent (every 10 sec) by the clusterheads to their neighbors, with a TTL = 2. A member which does not receive a maintenance notification message from its GCKS during 50 seconds, initiates a clustering process and invites its 2-hops neighbors to join its cluster. We note that, for implementation and applicability reasons, each cluster holds an IP multicast address selected by its GCKS.

The BALADE agent simulates the global controller, the local controllers and the group members. The global controller ensures the group key distribution to the local controllers, which forward it to their local member every 200 seconds. Like DMGSA, each cluster holds its IP multicast address. Group members join the multicast group formed by the local controller when they join its cluster. The clustering process, realized by OMCT, is implemented to ensure a highly correlated clusters, where local members are at only one hop from their local controller.

The Hi-GDH agent simulates a fixed number of multicast groups composed of 10 members in each scenario (varying from 3 to 5 corresponding to 30, 40 and 50 total members number). Each local chief generates and distributes its local key to its members every 200 seconds.

4 Results and analysis

We have realized simulations in order to evaluate the impact of the density of the group members in the network, the impact of the adhesion frequency to the multicast group, the impact of the mobility model and the impact of the relative speed on the performances of four group key management protocols, BALADE, DMGSA, GKMPAN and Hi-GDH. Each protocol belongs to one category of the taxonomy presented in section 2.

4.1 Impact of the density of the group members

The results of our simulations are presented in figures 3, 4 and 5. Figure 3 shows a comparison of the transmission key delay, for the four discussed group security protocols according to different densities of the group members in the network. Figure 4 shows the energy consumption of the discussed protocols, during the simulation. And finally, figure 5 presents the keys delivery rate (%) of the four approaches.

The BALADE group key management protocol brings a profit in the average delay of keys transmission, evaluated up to 30% compared to DMGSA and 90% compared to GKMPAN (cf Figure 3). This gain is due to the OMCT clustering process of BALADE, which ensures a highly correlated clusters. The keys transmission delay is than strongly dependent of the cohesion of the clusters around their local controllers. The efficiency of the key distribution process in BALADE is followed by a satisfying level of reliability (cf. Figure 5). BALADE brings a profit in term of keys delivery rate, varying from 20% to 40% compared to DMGSA for the high densities of members (in a surface of 500*500). Indeed, the keys distribution follows 1-hop ways between the clusterheads and their local members, thus minimizing the loss and interferences rate. Only the distribution of the group key follow a multi-hop ways between the global controller and the local controllers.
BALADE minimizes the nodes relays responsible for the keys forwarding ; the judicious choice of the local controller by their geographical localization induces an energy consumption optimization. However, the realized simulations show that the performances of BALADE are better when it is deployed with high densities of group members. These densities limit the inter-clusters communications and enhance the reliability, the delay and the rate of key transmission (cf. Figure 4).

Within DMGSA, the average delay of key transmission is directly affected by the 2-hops clusters formation. Indeed, in addition to the clusterheads which convey the group keys at 1 hop, the intermediary nodes are also responsible for routing these keys to the local members of the clusters, at 2 hops. The key transmission delay in DMGSA (cf. Figure 3) is thus definitively higher than that in BALADE. However, a considerable profit is noted compared to GKMPAN.

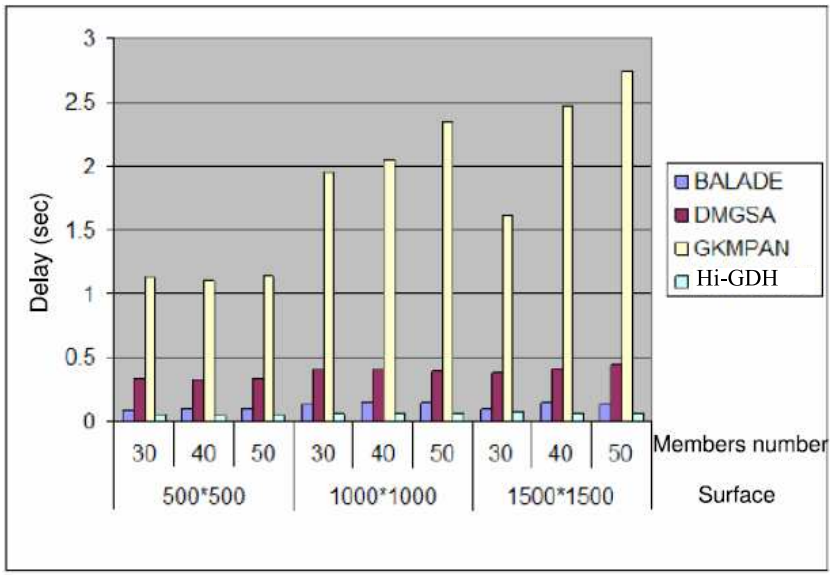

Fig. 3: Average delay of key transmission (seconds)

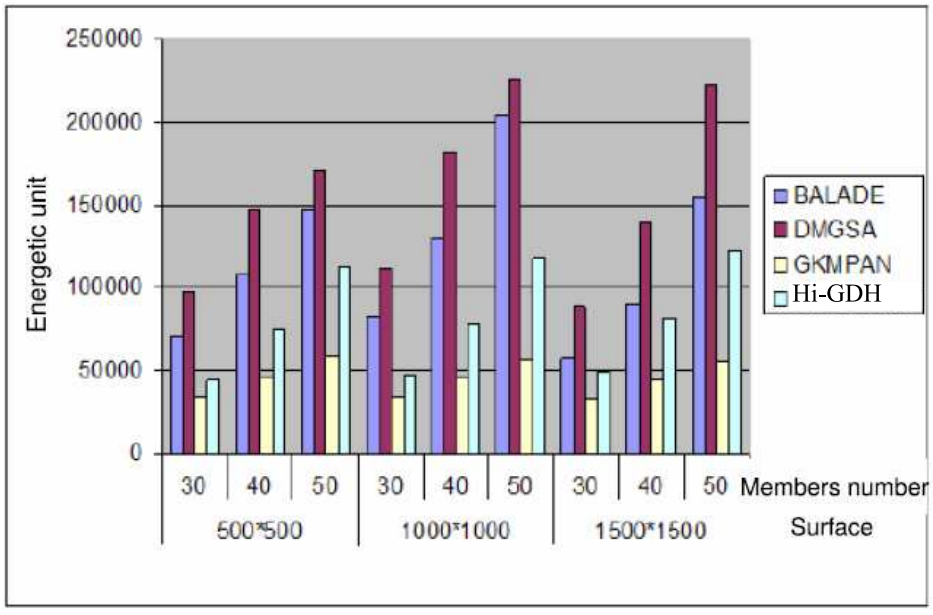

Fig. 4: Energy consumption

The energy consumption in DMGSA is the highest compared to BALADE, GKMPAN and Hi-GDH (cf. Figure 4), for all the configurations of densities of group members in the network. This is also explained by the significant number of relays which convey the keys to the members of the group.
DMGSA brings the best reliability rate of the keys transmission in the context of multicast groups with low densities (eg. 50 members in 1500*1500) (cf. Figure 5). This is explained by the fact that the 2-hops clustering processed within DMGSA allows a larger cover of nodes geographically far from the centre of the surface of the network.

In the GKMPAN protocol, only one entity is responsible for the distribution and the renewal of the group key to the members. The keys distribution in GKMPAN is secured by using intermediary operations of encryption and decryption of the group key by the nodes which forward keys to their neighbors. In GKMPAN, the keys transmission delay is evaluated as the sum of the delays necessary to the operations of encryption and decryption of the group key, by the intermediary nodes (hop-by-hop). This delay is consequently very high (cf. Figure 3) and makes difficult a fast accessibility of the data flow sent by the source. The synchronization between the source and the receivers for the encryption and the decryption of the multicast flow is not also guaranteed.
In term of energy consumption and reliability of the keys transmission (cf. Figure 4, 5), GKMPAN holds the best performances when it is deployed with high densities in the network, at the cost of a high delay of keys transmission.

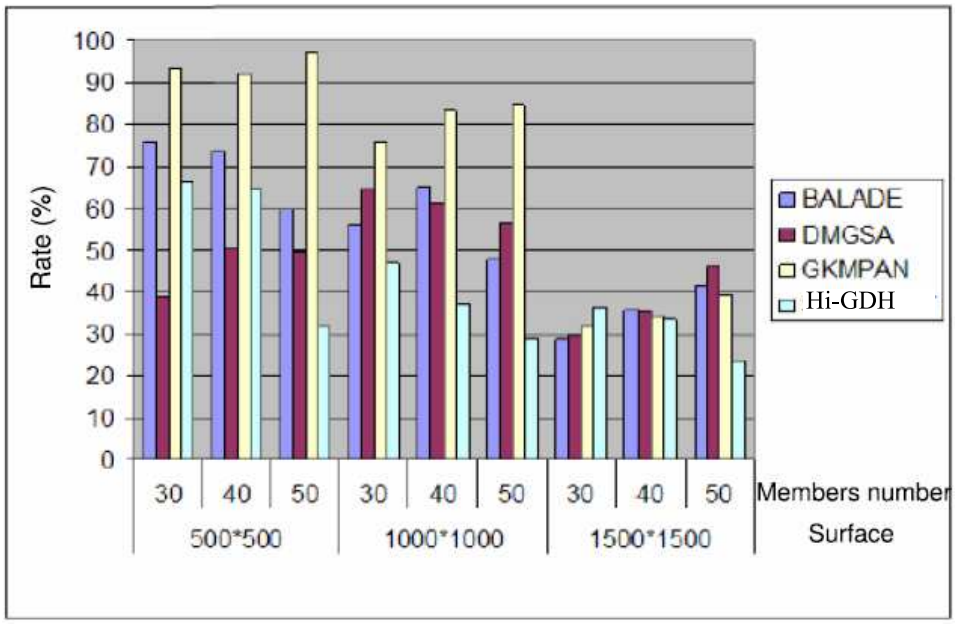

Fig. 5: Keys delivery rate (%)

The Hi-GDH protocol holds the best average delay of keys transmission (cf. Figure 3). This can be explained by the anticipation mechanism of the keys generation which reduces the encryption time of these keys. The use of only one message for the keys transmission to all the classes also reduces the average delay of keys transmission.
For the KDR metric (cf. Figure 5), we note that the performances of Hi-GDH increase when it is deployed within small densities of group members in the ad hoc network, where the probability of interferences of the communications is smaller.
The energy consumption of the Hi-GDH protocol is higher than the GKMPAN protocol, because of its

decentralized architecture, but lower than BALADE and DMGSA because of their clustering algorithms.

4.2 Impact of the adhesion frequency to the multicast group

We have demonstrated in [BCF05] that the adhesion frequency to the multicast group has an effect on the performance of the deployed group key management protocol according to its architecture. Indeed, we have compared in [BCF05] the behaviour of a centralized architecture (like GKMPAN) and a decentralized one (via the OMCT clustering algorithm used in BALADE). We have noted that for a centralized architecture, which suffers from the ”1 affects n” phenomenon, the keys renewal processes after each join or leave event has a negative impact on the average key transmission delay, the keys delivery rate and the energy consumption during the simulation.

In order to evaluate the impact of the adhesion frequency on the performances of the discussed group key management protocols, we simulate them under two configurations. The first one corresponds to a static adhesion configuration, in which all nodes of the network are members of the multicast group from the beginning of the simulation. In the second configuration, the adhesion to the multicast group is dynamic and each Join or Leave event is followed by a keys renewal process triggered every 5 seconds by the controllers of the group key management protocols (the group membership changes are random and the keys renewal is carried out every 5 seconds). We define within these two configurations the number of members (=40) and the surface (=1000*1000m), which represents the average density of the simulations presented below. The results of these simulations are given in Figure 6.

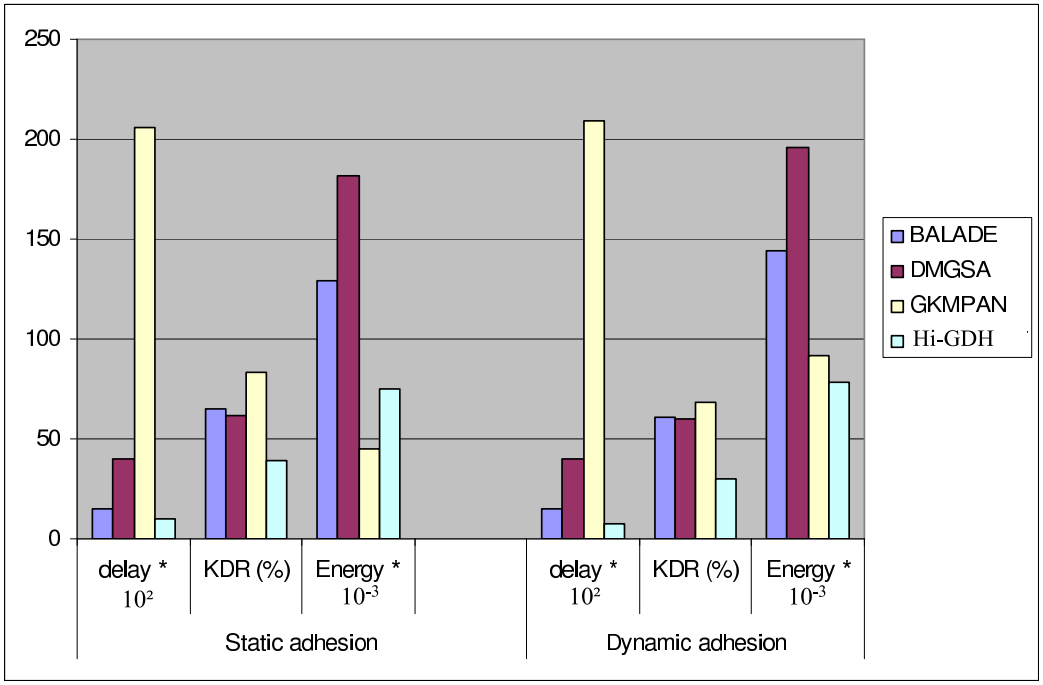

Fig. 6: Impact of the adhesion frequency on the GKMPs

Concerning the energy consumption, Figure 6 shows that the performances of the protocols BALADE, DMGSA and Hi-GDH are almost the same for the static and dynamic adhesion configurations. On the other hand, the consumption of GKMPAN in term of energy doubles from the static adhesion configuration to the dynamic one. We can explain this difference by the fact that the keys renewal processes executed after each join or leave event affects all the group members in the case of GKMPAN and only the affected clusters or classes in the case of the others protocols. The GKMPAN protocol suffering from the ”1 affects

n" phenomenon because of its centralized architecture, is so the most dependant of the adhesion frequency to the multicast group.

For the average key transmission delay and the keys delivery rate metrics, all the simulated group key management protocols have the same performances with the two adhesion configurations. We note however that the keys delivery rate in Hi-GDH decreases considerably with the high frequency adhesion configuration, due to the increase of the interferences.

4.3 Impact of the mobility model

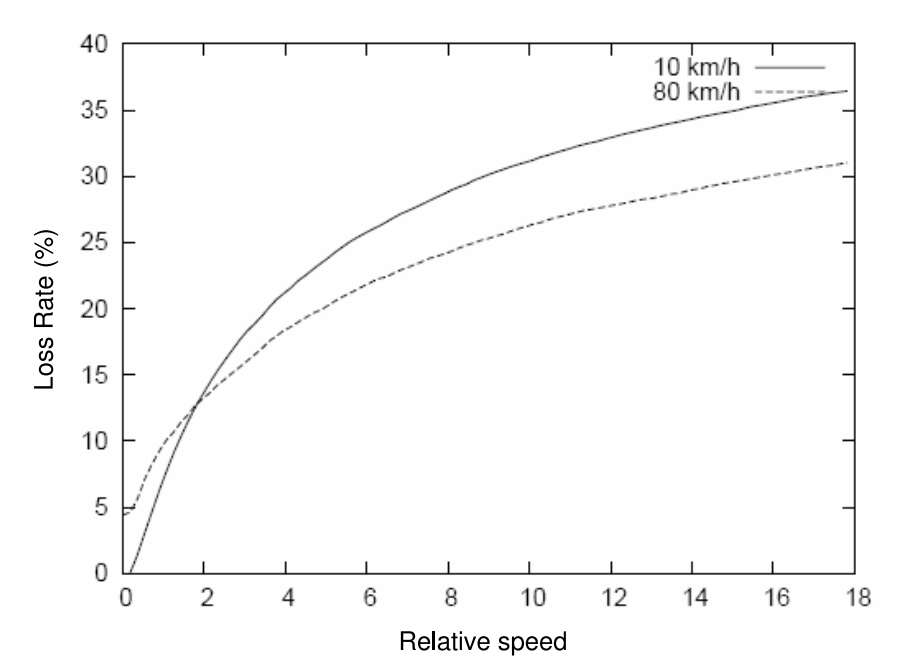

Fig. 7: Impact of the relative speed on the loss rate

We have shown in [BCF06b] that the model of nodes mobility (individual or group mobility) has a direct impact on the performances of the deployed group key management protocol, in terms of average delay of keys transmissions, energy consumption and keys delivery rate. These group mobility models we studied are the RWG (Random Waipoint Group), the MHG (Manhattan Group Mobility) and the SQG (Sequential Group Mobility), provided by the *grcmob* group mobility generator.

The movement of groups is characterized by several parameters, such as its speed. Within the same group, the speeds of its members are not identical, they take their values within an interval around the speed of the group. The length of this interval is called the relative speed, implemented in *grcmob* [ZK04]. We have realized simulations to evaluate the effect of the relative speed parameter on the loss rate of the transmitted keys in Hi-GDH. We chose this protocol because it is the most suitable to use a group mobility model. The scenario that we simulate is the TEK distribution sub-protocol presented in 2.3, for only one group with varying number of members, for the two speeds of the group (10 km/h and 80 km/h). The relative speed starts with 0 km/h (all group members have the same speed) to 18km/h. Figure 7 shows the impact of the relative speed on the loss rate of the sub-protocol.

These results demonstrate a high correlation between the relative speed and the loss rate with Hi-GDH. The rate of lost packets increases when the relative speed between group members increase too. We note

also that the loss rate is higher in the case of group movement speed equal to 10km/h. This can be explained by the fact that the ratio between the relative speed and the group movement speed is much higher with small speeds of groups (18/10 compared to 18/80) ; thus more affects the reliability of communications.

5 Conclusion

In this paper, we showed the importance and the efficiency of performances evaluation of group key management protocols within ad hoc networks. Indeed, it allows to demonstrate the effective applicability and adaptation of the discussed protocols to the context of MANETs.

We presented in this context our method to evaluate the performances of group key management protocols within MANETs, based on the NS2 simulator. We define in a first step the parameters and the metrics of our evaluation. Then, we illustrate our approach by evaluating some existing group key management protocols within MANETs: GKMPAN, DMGSA, BALADE and Hi-GDH. We have also evaluated the impact of the adhesion frequency to the multicast group, the impact of the mobility model and the impact of the relative speed on the performances of these protocols in terms of average delay, reliability and energy consumption of keys delivery processes. We conclude from our simulations that the choice of the group key management protocol within MANETs is highly dependent of two main constraints: the security policies defined by the group communications application and the quality of service to provide to the group members. For example, for a large multicast group deployed within ad hoc networks whith high dynamicity and mobility of nodes, BALADE and DMGSA are the most suitable GKMPs, which provide the best compromise in terms of average delay of keys transmissions, energy consumption and keys delivery rate. For a small multicast group where the dynamicity of nodes is limited, GKMPAN is the most adapted because its avoids the overhead due to the clustering process while providing acceptable performances in terms of average delay and keys delivery rate. Finally, for an hierarchical multicast group where members are belonging to classes with different levels, the use of the Hi-GDH protocol is the most suitable to take into account the characteristics of such applications.

References

[BCF04] M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad Hoc Networks. In *Networking 2004, Third International IFIP TC6 Networking Conference*, volume 3042 of *LNCS*, pages 725–742, Athens, Greece, May 9-14 2004. Springer.

[BCF05] M.S. Bouassida, I. Chrisment, and O. Festor. Efficient Clustering for Multicast Key Distribution in MANETs. In *Networking 2005, International IFIP TC6 Networking Conference*, volume 3462 of *LNCS*, pages 138–153, Waterloo, CANADA, May 2005. Springer.

[BCF06a] M.S. Bouassida, I. Chrisment, and O. Festor. Group Key Management in MANETs. Accepted for publication. *International Journal of Network Security IJNS*, 2006.

[BCF06b] M.S. Bouassida, I. Chrisment, and O. Festor. Mobility-Awareness in Group Key Management Protocols within MANETs. *Annals of Telecommunications*, 61(9-10), 2006.

[CH03] T. Chiang and Y. Huang. Group Keys and the Multicast Security in Ad Hoc Networks. In *Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPP Workshops)*, page 385, 2003.

[DH76] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[KLG06] J. Kong, Y. Lee, and M. Gerla. Distributed Multicast Group Security Architecture for Mobile Ad-hoc Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, Nevada, USA, April 2006.

[KLNY03] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz. Secure Multicast Groups on Ad hoc Networks. In *Proceedings of the 1st ACM workshop on security of ad hoc and sensor networks*, pages 94–102. ACM Press, 2003.

[LP03] L. Lazos and R. Poovendram. Energy-Aware Secure Multicast Communication in Ad Hoc Networks Using Geographical Location Information. In *IEEE International Conference on Acoustics Speech and Signal Processing*, pages 201–204, 2003.

[MME04] M. Moharrun, R. Mukkalamala, and M. Eltoweissy. CKDS: An Efficient Combinatorial Key Distribution Scheme for Wireless Ad Hoc Networks. In *IEEE International Conference on Performance, Computing and Communications (IPCCC)*, Arizona, April 2004.

[ZK04] Y. Zhu and T. Kunz. MAODV Implementation for NS-2.26 - Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01, January 2004.

[ZSXJ04] S. Zhu, S. Setia, S. Xu, and S. Jajodia. GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks. In *MobiQuitous*, pages 42–51, 2004.